Machine Learning Nanodegree
Capstone Project Proposal

Named Entity Recognition

Nitin Bhandari
Feb 23, 2018

1 Domain Background

Named Entity Recognition (NER) also known as entity identification, entity chunking, and entity extraction is an important sub task of information extraction that locates and classifies named entities in text into pre-defined categories such as name of the person, location, expressions of time, monetary values, etc. [1]

For example:

Elon deleted SpaceX page on Facebook on March 23, 2018.

And the system produces an annotated block of text that shows the name of entities:

[Elon]$_{\text{Person}}$ deleted [SpaceX]$_{\text{Organization}}$ page on [Facebook]$_{\text{Organization}}$ on [March 23, 2018]$_{\text{Time}}$

1.1 Motivation

The first time I was introduced to machine learning, I learnt about Natural Language Processing and did an introductory project on named entity recognition using Naïve Bayes. So, for the capstone project, I choose the same domain to build an efficient system.

2 Problem Statement

NER is a subtask of information extraction. While the expression 'named entity' may seem the task to those entities such as word or phrases, stands consistently for a particular reference like those designated as rigid designators but is in fact not strict and can be used for temporal expressions*.

For example:

***Temporal annotation** is the study of how to automatically add semantic information regarding time to natural language documents. It plays a role in natural language processing and computational linguistics.

In sentence ‘The automotive company created by Henry Ford in 1903’, Ford refers to the Ford Motor Company although there are other meanings of Ford as well.

‘The year 2001’ and ‘I take my vacations in June’, the first phrase refers to 2001 year of the Georgian calendar and the second phrase refers to the month June of unknown year.

The problem of NER is often broken down into two distinct problems:

a) Detection of named entities (segmentation) and
b) Classification of the name by the type of entity they refer to (Ontology)

3 Dataset and Input

The NLTK python library does not have a standard corpus for NER in English. But it does have the famous CoNLL 2002 Named Entity CoNLL but in Spanish and Dutch. We will be using dataset provided on the Kaggle annotated with IOB and POS tags [4]. The dataset can be downloaded from here.

4 Solution Statement

The model relies heavily on good annotated corpus for the system to learn and produce great results. We are going to implement a basic CRF system for NER and modify the parameters to get a better result.

5 Benchmark Model

Since the NER is notoriously challenging problem in the world of information extraction because of different languages, wide variety of datasets, to name a few areas that widely impacts the performance of the NER. Often, Stanford’s Named Entity Recognizer is often considered as the benchmark model that uses Conditional Random Fields (CRF). With the advancements in Neural Network this paper claims Bidirectional Jordan RNN outperforms the CRF-baseline by 14% in relative error reduction [3].

6 Evaluation Metrics

The project will be using precision, recall and F1 score to check the performance of the system.

7 Project Design

7.1 Programming Language and Libraries

The project requires:

- Language : Python
- Libraries : pandas, numPy, sklearn, elif5

7.2 Machine Learning Design

7.2.1 Load the data into DataFrame using pandas
7.2.2 Retrieve sentences along with their labels
7.2.3 Prepare the dataset
7.2.4 Implement CRF using sklearn-crfsuite
7.2.5 Perform cross-validation
7.2.6 Get the precision, recall and F1 score using scikit-learn classification report
7.2.7 Use elif5 library to visualize transition probabilities from one tag to another
7.2.8 Parameter Tuning
7.2.9 Go to step 6 and repeat step 8 and step 6 till you achieve a higher accuracy

References:

[1] https://en.wikipedia.org/wiki/Named-entity_recognition
[2] https://nlpforhackers.io/named-entity-extraction
[3] http://www.iro.umontreal.ca/~lisa/pointeurs/RNNSpokenLanguage2013.pdf
[4] https://www.depends-on-the-definition.com/named-entity-recognition-conditional-random-fields-python
[5] https://en.wikipedia.org/wiki/Conditional_random_field